1. Тема типового расчета

Тема — база данных для работы с сервисом для онлайн-школы программирования, который будет давать возможность ученикам выбирать себе курсы, а также осуществлять удобное взаимодействие учеников и преподавателей.

1. Описание бизнес-процесса

Требуется создать базу данных для работы с сервисом для онлайн-школы программирования, который будет давать возможность ученикам выбирать себе курсы, а также осуществлять удобное взаимодействие учеников и преподавателей.

База данных будет включать в себя информацию о курсах, учениках, которые проходят этот курс, а также данные о преподавателях, ведущих онлайн-занятие. Каждый пользователь имеет свои возможности и доступы к данным.

Пользователь должен иметь возможность видеть информацию о курсах, которые ведутся в онлайн-школе.

Выбрав себе курс, пользователь переходит в раздел ученики. Каждый ученик определяется в учебную группу и имеет доступ к купленному курсу, наличие информации о преподавателе и его контакты, а также наличие теоретических материалов по приобретенному курсу. Ученик имеет личный список домашних работ, он вносит ответы на задачи и получает оценку. По окончании курса ученику выдается сертификат о его прохождении если сделаны все домашние работы, в противном случае получение сертификата откладывается до того момента, пока все задания не будут выполнены.

Преподаватель имеет информацию о своих учениках – список учебных групп. Преподаватель может взять себе ещё одну или несколько групп, а также отказаться от какого-то количества. Также преподаватель имеет доступ к домашним заданиям учеников, он может добавлять, удалять и просматривать выполнение работы. Каждую домашнюю работу ученика преподаватель оценивает (она может быть просто не сделана или же сделана неправильно) и вносит эти данные, которые видят ученики и могут определить, нужно ли переделать работу.

Администратор имеет доступ ко всему. Он может принять ученика на курс или исключить его. Принять или уволить преподавателя, а так же имеет все те же доступы, что и у преподавателя.

1. Категории пользователей

Для данной БД должны быть предусмотрены 4 группы пользователей:

* Администратор — должны иметь возможность принимать и исключать учеников, принимать и увольнять преподавателей, также имеет доступ ко всей информации, что есть у преподавателя.
* Преподаватель — должны иметь доступ к списку своих учеников и их домашним работам. Может установить/изменить/удалить статус (оценку) домашней работы, имеет возможность отказаться от группы учеников или взять новую;
* Ученик — должны иметь информацию о своем преподавателе и номере учебной группы, описание курса, а также список домашних работ с теоретическим материалом. Могут добавлять и удалять выполненную домашнюю работу;
* Пользователь – должны иметь возможность просматривать информацию о курсах, которые есть в онлайн-школе

1. Концептуальное проектирование. ER-диаграмма. Таблицы, индексы, ключи, связи, ограничения

Все данные, которые должна содержать БД представим в виде таблиц.

1. Таблица User (информация о пользователях сервиса)

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Пояснение |
| **User\_id** |  | Id пользователя |
| Usergroup\_id |  | Id группы пользователей |
| login |  | логин |
| password |  | пароль |
| name |  | имя |
| lastname |  | фамилия |
| Date\_of\_birth |  | Дата рождения |
| number |  | Номер телефона |

1. Таблица UserGroup (определение категории пользователя)

Пример: 1-пользователь

2- ученик

3-учитель

4- администратор

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Пояснение |
| Usergroup\_id |  | Id группы пользователей |
| Usergroupname |  | Наименование группы пользователей |

1. Studygroup\_user (таблица для идентификации пользователя как ученика – определяем в учебную группу)

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Пояснение |
| User\_id |  | Id пользователя (только ученики) |
| Studygroup\_id |  | Id учебной группы |

1. StudyGroup (таблица с группами студентов, уточнение: разные группы могут изучать один и тот же курс)

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Пояснение |
| Studygroup\_id |  | Id учебной группы |
| Study\_id |  | Id курса |
| Teacher\_id |  | Id учителя |

1. Study (таблица для получения описания курса)

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Пояснение |
| Study\_id |  | Id курса |
| Description |  | Описание курса |

1. Homework (таблица с описанием домашних работ, уточнение: домашняя работа одинакова для учеников одного и того же курса)

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Пояснение |
| Homework\_id |  | Id домашнего задания |
| Teory |  | Теория |
| Study\_id |  | Id курса |
| Description |  | описание |

1. Homework\_User (таблица с домашними работами студентов, уточнение: каждый студент выполняет домашнюю работу, вносит полученный ответ, а преподаватель ставит оценку)

|  |  |  |
| --- | --- | --- |
| Имя поля | Тип данных | Пояснение |
| Homework\_id |  | Id домашнего задания |
| id |  | Id записи |
| User\_id |  | Id пользователя (ученика) |
| answer |  | ответ |
| mark |  | оценка |

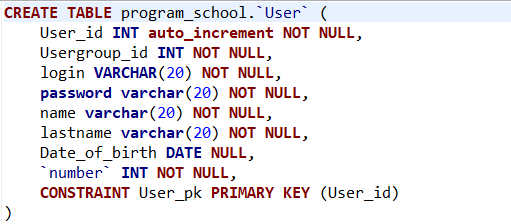
Ограничения:

1-Пользователь определен однозначно: пользователь не может быть учителем, учеником или администратором одновременно, ученик не может быть учителем, администратором, пользователем и тд.

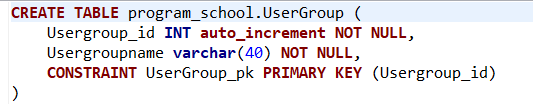
2-Один ученик определен в одну учебную группу, ученик не может находиться в разных учебных группах.

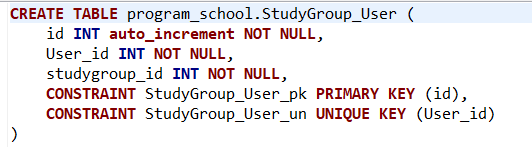
3-Учитель может преподавать в разных учебных группах, при этом одна учебная группа закреплена за конкретным учителем.

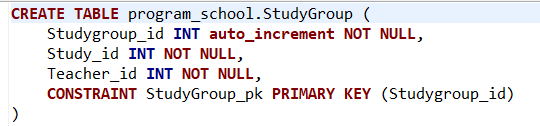
4-Одна домашняя работа распространяется на всех учеников всех групп одного и того же курса, при этом каждый ученик вносит свой ответ и получает свою оценку.

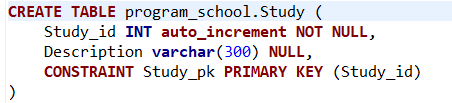


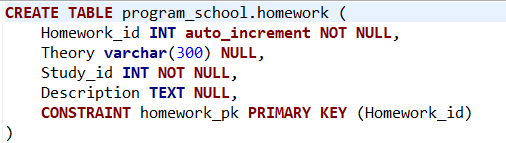


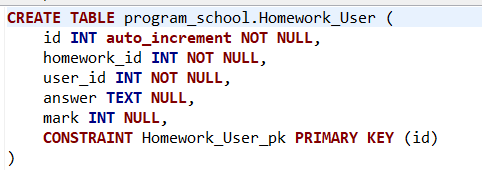




















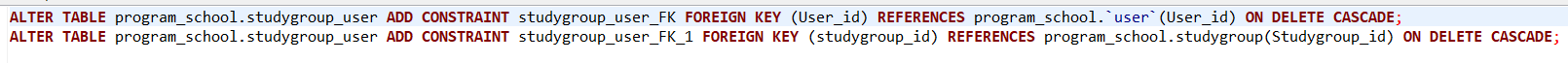


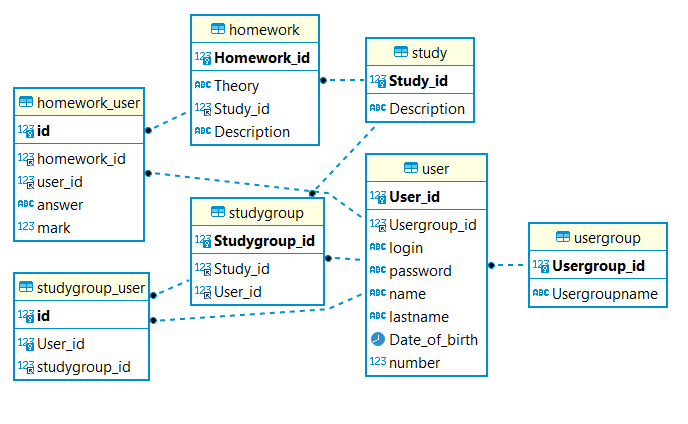












**CREATE** **FUNCTION** program\_school.view\_students(studygroup\_id **int**)

**RETURNS** **table**(

"Course\_description" **text**,

"FIO" **varchar**(100),

"All\_hw" **int**,

"Done\_hw" **int**,

"Average\_mark" **float**

) **as** $$

**begin**

**select** st.description **as** "Course\_description",

**concat**(u.name, ' ', u.lastname) **as** "FIO",

**count**(hu.homework\_id) **over** (**partition** **by** u.User\_id) **as** "All\_hw",

**count**(**case** **when** hu.mark >= 3 **then** 1) **over** (**partition** **by** u.User\_id) **as** "Done\_hw",

**avg**(hu.mark) **over** (**partition** **by** u.User\_id) **as** "Average\_mark"

**from** studygroup **as** sg

**inner** **join** study **as** s **on** sg.study\_id = s.study\_id

**inner** **join** studygroup\_user **as** su **on** su.studygroup\_id = sg.Studygroup\_id

**inner** **join** 'user' **as** u **on** su.User\_id = u.User\_id

**inner** **join** homework\_user **as** hu **on** u.User\_id=hu.user\_id

**where** su.studygroup\_id = studygroup\_id

**end**;

$$ **language** plpgsql;

